An Analog History of Procedural Content Generation

Gillian Smith
Northeastern University, Playable Innovative Technologies Lab
360 Huntington Ave, 100 ME
Boston, MA 02115, USA
gi.smith@neu.edu

ABSTRACT

Procedural content generation (PCG) is typically considered a feature of digital games. Commonly cited "first" uses of PCG are usually digital games from the early 1980s: Rogue or Elite. However, when broadly construed, PCG simply means that content is generated following a formal procedure—the agent that enacts this procedure is merely assumed to be a computer. The precursors to what we now call PCG are to be found in games where it is a human who is asked to follow a procedure to generate game content, and the earliest *digital* uses are crude reproductions of those same games. This paper explores the role that PCG has played in analog games and how it has influenced PCG in digital games. In doing so, we can gain insight into the nature of content generation and can more easily define the boundaries of what we typically consider PCG to encompass.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence]** Applications and Expert Systems – *Games*. K.8.0 [**Personal Computing]** General – *Games*.

General Terms

Design.

Keywords

Procedural content generation, game history, non-digital games.

1. INTRODUCTION

Procedural Content Generation (PCG) is use of a formal algorithm to generate game content that would typically be produced by a human. Recent years have seen research into methods for producing a wide variety of such content, including levels, weapons, textures and procedural effects, and story content [18]. Methods for generating such content range from relatively rule-based systems to search-heavy processes [33,43]. Commonly cited motivations are to use PCG to improve game replayability, reduce a perceived authoring burden, or introduce adaptive and personalized content [34].

Within games academia, PCG research is predominantly rooted in the sub-field of game artificial intelligence, where it is framed as an artificial intelligence problem for how to replicate or replace the creativity and design intent of a professional human designer, how to have an AI reason about the quality of the content it has created, and how to build an AI system that can interact with another human designer as a creative partner [22,32,35]. Thus, PCG also overlaps with game design: whether intentional or not, every PCG system implicitly encodes a formal theory for both the game design process and the product that is being procedurally created.

By casting PCG as a problem of formalizing a design process, rather than a deeply technical artificial intelligence problem, we can view PCG more abstractly, and look at a more broad definition of PCG than that of a computer creating content for a game. We can look to other generative formalizations of design and draw parallels between the PCG we recognize in modern computer and video games and generative processes in other kinds of games, including board games, card games, and tabletop roleplaying games. Taking this broader perspective on PCG permits a closer analysis of the kinds of formal theories encoded in their generative systems—in board games, the rules for generation are explicitly laid out for players to be able to follow, as opposed to computer games where the rules are locked up in code. It also allows us to gain a deeper understanding of the nature of PCG, better describe the motivations for using PCG in games, and see how the role and function of PCG in games has, or has not, changed over time.

This paper examines the analog roots of procedural content generation: the use of randomness and how it can be directed, the careful design of modular content, and the creation of formal game systems that aid player creativity. These precursors greatly influenced the first digital games to use PCG, with the earliest occurrences of PCG being direct copies of their non-digital ancestors. Looking at this progression, it is possible to see a general trend for using data-heavy, knowledge representation-dependent approaches that is now being challenged by algorithm-intensive models of design and creativity. Examining non-digital approaches to PCG also reveals that the motivations for why content generation is desirable in games has gone largely unchanged across the decades and across media: there are common themes in a desire for replayability: for surprising the player, and for providing variety.

2. RELATED WORK

Existing work in understanding PCG has largely come from the technical games research community. Hendrikx et al. survey the range of content that can be produced and the common algorithms used to produce it, but focus entirely on digital games [18]. Togelius et al.'s survey on "search-based" procedural content generation [43] covers the many ways in which genetic algorithms and other optimization approaches have been used to generate content for games. This taxonomy draws distinctions between the kind of content being produced, the necessity for the content in overall gameplay, and the technical approach used. Of particular interest to this work, however, is the observation that PCG does not need to be stochastic: deterministic content generation has been used for programmatically creating universities (such as in *Elite* [6] or *No Man's Sky* [17]) with no randomness. This paper

Proceedings of the 10th International Conference on the Foundations of Digital Games (FDG 2015), June 22-25, 2015, Pacific Grove, CA, USA. ISBN 978-0-9913982-4-9. Copyright held by author(s).

focuses almost entirely on PCG that incorporates a random element.

Smith's taxonomy for PCG breaks down the design elements of games and PCG systems using the MDA framework [34]. The focus of this paper is mostly on the mechanics of PCG systems, specifically focused on the data representation used and the simple, constructive algorithms found in non-digital games. This paper also presents several motivations for the use of PCG in analog games and its transition to digital games.

Togelius et al. make a distinction between user-created content and procedural content generation [42], stating that though they share some common output, the fact that the content is created by human players as opposed to a computer means that it is not procedurally generated. This lies directly at odds with the claims of this paper: procedurally generated content can be created by a human, if that human is following a clearly defined procedure. There is an important distinction between player-created content where the player has complete freedom in what they make (e.g. *Pictionary* [16]) and one where they are constrained by a system of rules (e.g. *Carcasonne* [49]).

3. DIRECTING RANDOMNESS

Randomness is a hallmark of procedural content generation. Though there exist games that incorporate deterministic PCG, the majority include some kind of random decision-making, even if the deterministic nature comes from holding the random seed constant or controllable. Prominent roguelike designer Andrew Doull even includes randomness in his definition of PCG [9].

Game designer Dan Kline has argued that a key to good game design with procedural systems is to replace uniform randomness with directedness [21]. This directedness can take many forms, however: directing random content creation can occur through carefully designing the individual pieces of content being assembled or through altering the algorithm for assembly. This section provides examples of different ways in which this can be done, motivated by examples from pre-computer games.

3.1 Modularity in Design

One of the important tasks for a PCG system is to ensure that the content it creates is "playable"—at a bare minimum (though often this minimum is considered sufficient) the content made by the system should be valid, meeting the rules of the game and being sufficient for a player to use it in the game. Playability of content in games where it is assembled randomly relies upon heavily modular design: each piece should be able to fit with every other piece. This kind of modular design is common in non-digital games.

In 1976, TSR Hobbies released a set of "Dungeon geomorphs" as a playing aid for *Dungeons & Dragons*. These were a set of 15 tiles—10 large and 5 small—that could be combined together in many different ways in order to produce dungeon layouts: TSR advertised that they would create an "endless number" of dungeons [45]. The tiles are each authored to contain numerous passageways, doors, and walls, and designed so that they can fit next to each other in any combination to produce a valid dungeon. The instructions for use suggest that the player assembling the dungeon, presumably a dungeon master using the tool as a design aid, then annotate the dungeon layout with monsters, objects, or traps, as well as modify connections between tiles if desired to alter how the player can move around the generated environment.

In designing *Simon* for Milton Bradley in 1978 [25], Ralph Baer adopted the technique of carefully designing modular pieces that could be randomly chained. *Simon* is a game that produces randomly ordered sounds and asks the player to repeat them back in the same order. An earlier 1974 Atari game, *Touch Me [3]*, had the same core mechanic. Baer had played *Touch Me* at a trade show in 1976, before *Simon* was conceived. *Touch Me* suffered the problem that its casing was not attractive and the sounds it produced were, as Baer described, "non-sonorous" [53]. In the design of *Simon*, Baer copied all of *Simon* from *Touch Me*, but with one important difference: he deliberately crafted the sounds that *Simon* would make to sound good when played in any order, based on bugle notes. This authorship, though simple, of the notes that are then pieced together entirely randomly results in a far different experience for the player than *Touch Me*, where the same random process is followed, but the design of the notes themselves is not directed.

These kinds of directing or authoring choices are echoed in modern, digital PCG, where building blocks and the transitions between them are explicitly designed to work well together. For example, *Spelunky*'s level generation system [20] has pre-authored tiles are placed next to each other, then decorated with enemies and loot in a second pass. This kind of simple construction algorithm (though a human may choose to use their creative capacity to take a more sophisticated approach than pure randomness) is only possible due to the careful design of the underlying building blocks for the system.

The underpinnings of modular design in analog games are rooted far earlier than the late 1970s. Dice, tiles, and cards are all designed as modular game platforms, in which many games can be played. It is typical for games using these pieces to involve random generation of a tableau, even creating randomized physical spaces for players to navigate, such as mazes [30]. The kind of tableau generated can be influenced by selecting which distribution of game pieces should be drawn.

3.2 Algorithmic Assembly

Instead of (or in addition to) the careful crafting of building blocks to work together nicely, a second method for directing randomness is to layer complexity into the randomly-driven algorithm used to create the content. Analog games, in their need to explicitly describe the construction algorithm in order that a human can follow it, make it possible to see patterns in these constructive approaches. Interestingly, the design of this algorithm also may be constrained by a requirement that computer implementations of PCG do not need to adhere to: that the act of constructing the content be part of a playful experience.

Perhaps the most influential example of guided randomness for content generation in an analog game is the random encounter and random dungeon creation guidelines for *Dungeons & Dragons* (D&D) [13,15]. The original wilderness encounter guidelines were heavily based on those from Avalon Hill's *Outdoor Survival* [10]: a single dice roll would determine first whether the player is lost or has an encounter, then another to determine the type of encounter, and finally a third to determine what specific kind of creature the encounter is with. Each of these rolls is further modified by the type of terrain the player is in. The *Advanced Dungeons & Dragons* (AD&D) guide also provides instructions for using a similar lookup table model for generating entire dungeons, with each die roll corresponding to the type and position of room, passage, or doorway. These guidelines originally appeared as a supplement authored by Gygax for the original D&D system [13,14].

The use of a multi-layered lookup table for directed randomness is easy to extend, with many modules and articles from fan magazines offering new elaborate systems built upon this model (e.g. [23,26]). Each table offers a new way to customize the content being generated based on current player choice or game state. However, even with this kind of direction, the dungeon master (DM) is still cautioned in the official game rules to exert directorial control over encounters, and to use their own judgment when incorporating procedurally generated content or encounters:

> *A common mistake most DMs make is to rely too much on random die rolls ... The DM must use good judgment in addition to random tables.* [8:X59]

Similarly, the AD&D random dungeon generator cautions DMs that "discretion must prevail at all times" [13:3] including requiring that the human executing the procedure decide how to handle any conflicting die rolls that would result in an invalid dungeon layout.

Lookup tables have also been used to generate stories, where there is less randomness involved and more reliance on player choice to determine which lookup table and what entry to use for generating content. For example, the board game *Tales of the Arabian Nights* [50] has players follow a system of dice rolls, modulated by player choices, to look up passages in a large book in order to assemble individual stories.

Replacing or augmenting the random number generation with player choice is a common method for generating game spaces as part of the game process. Tile-based games like *Carcassonne* [49] and *Betrayal at House on the Hill* [4] share much in common with procedural generation of game spaces due to the organic unfolding of the game space during play, following a procedure that is co-determined by players. *Betrayal at House on the Hill* has players collaboratively and procedurally generate the mansion while they are exploring, following game rules for where tiles are allowed to be placed and the floor that the tiles are allowed to be placed on. Each player plays a single tile on their turn as they work together to explore a haunted mansion. This kind of multiplayer co-creation of space, guided by a system, is unexplored in digital games.

4. THE TRANSITION TO DIGITAL

Game designers and toy manufacturers adopt new technologies quickly: almost as soon as there existed a computer, there existed a game that could be played on it [19]. Some of the earliest uses of PCG in digital games were direct copies of the PCG systems found in analog games of the time period. This transition from analog to digital PCG can be seen clearly in two different domains: the hobbyist community that sprung up around the personal computer and the incorporation of computing in tabletop roleplaying games.

4.1 Hobbyist PCG

The rise of the personal computer and hobbyist programming led to a surge of interest in creative computing and hobbyist game programming. Hobbyist magazines of the time, such as *BYTE* magazine and *Creative Computing*, would publish programs for their readers to implement; these programs frequently involved some form of procedural content generation or generative design in non-game domains. Examples of programs include a 1976 interactive random haiku generator that uses random numbers to look up from lists of words and then puts them together following syllabic constraints [12] and a 1979 random music generator that could create four-part harmonies [2]. These simple generative systems typically used slightly directed randomness to create their artifacts, in one of the manners described in the previous section.

4.2 Tabletop Roleplaying

Perhaps the most commonly implemented early digital generative systems were based upon tabletop roleplaying, especially *Dungeons & Dragons* and *Tunnels & Trolls* [40]. Given the large amount of randomness used in generating all aspects of a roleplaying campaign, as well as the highly systemic nature of the game's underlying design, tabletop roleplaying systems lent themselves well to digital implementation.

One phenomenon in the early days of computing was the procedural generation of monsters and environments intended for use in a physical, tabletop roleplaying game. Three player-aid supplements for the *Runequest* roleplaying game [46,47,48] consisted entirely of monsters with computer-generated statistics (i.e. hit points, traits, spells, and abilities) ready for use in a tabletop campaign. These monsters were generated from a computer implementation of the tables typically used by DMs for generating characters. On a larger scale, Strategic Simulations Inc. released the *AD&D Dungeon Master's Assistant* program, which was intended for DMs to use on the personal computers during the campaign itself. This program would do all the dice-rolling upkeep typically performed by a DM, including generating monsters and encounters [52]. Again, this program was a direct implementation of the mathematical rules and random encounter tables found in the source material—this early use of digital PCG is a direct copy of its analog predecessor.

There were also early experiments with the procedural generation of entire game scenarios that would be played as a physical game. *A Computer Generated Dungeon* [51] was a computer-generated solo dungeon adventure for the *Tunnels & Trolls* system. Available for only a short period from 1977-1978, this solo dungeon followed the tropes and formulae for other, human-authored solo dungeons: a choose-your-own-adventure style set of rooms and encounters written as indexed passages on different pages, where the transitions between each passage were controlled by choices made by the player and combat resolved using the *Tunnels & Trolls* core rules. The passages in *A Computer Generated Dungeon* were clearly written by a human author, with solid English prose, but re-ordered by the algorithm to create a variety of different experiences, each sold for $5 with the promise as well of player customization: players could select the name of the monster and customize the text in the footer of each of the 21 pages in the booklet.

Finally, there were many attempts at producing entire recreations of roleplaying games, including the PCG portions of them, for playing on the computer. Most of these focused on building a single-player experience from a traditionally multi-player, social game. *Rogue* [44] and other early procedurally generated dungeon crawling games clearly were influenced by the systems in tabletop roleplaying. *Adventure Construction Set* offered the ability for players to create their own *Adventure*-style [37] games. To ease the process of developing the content for these games, the system included an auto-complete feature for level creation. If the player wanted to guide only a small part of the world-building process, they could offload the remainder of the effort to the computer. The player would be responsible for building important rooms and outdoor areas, and then could instruct the computer to finish the level, along with parameters for how it was allowed to perform this task and how destructive the system was allowed to be of human-created geometry. The system shared many of the same motivations as current work in mixed-initiative level generation.

5. EARLY PCG MOTIVATIONS

Examination of the marketing material and game reviews surrounding analog games with PCG shows a set of motivations that are almost identical to those used for contemporary PCG research and practice. The goals of providing a replayable experience and assisting players in their creativity are common framings for a large amount of PCG research [34]. We can also see a theme of hobbyists and artists using PCG as an expressive medium, both in analog and early digital PCG.

For early digital PCG, where the computer is generating content for an analog game or is directly copied from one, two new motivations emerge: the role of the computer as an unbiased agent, and of the computer's ability to make a traditionally multiplayer game accessible to individuals to play.

5.1 Replay

Especially for analog strategic or puzzle games that employ PCG, replayability is a common motivation. *Quantum* is an example of such a game—a strategy board game where the beginning configuration of pieces is completely randomized, and any configuration of initial puzzle pieces is valid. Advertisements for the game described the random distribution of start pieces as bringing "brand new fun" to playing [41], and the designer of the game, H. Peter Aleff, argues that the random beginnings means that "there is no way to bone up on standard openings as in chess" [7]. The game offers a way for players to practice strategies in different and unpredictable environments on each play.

Replay is also offered as a commodity to players: a promise that, with the purchase of or other investment in a single game, the player will have access to more content than the game appears to offer. *Cross Currents*, designed by prolific US board game designer Sid Sackson, uses dominoes to lay out a random board at the beginning of each game that players then traverse following mathematical rules that use the number of pips on each "tile". Sackson pitches the game to prospective players as a novel way to get a great deal of use out of a standard set of dominoes [30].

5.2 Assisting Creativity

Within tabletop roleplaying, PCG is primarily used for assisting the DM in their creativity. With the DM needing to serve as an on-demand game designer, managing and creating content for their players who have the ability to engage in hard-to-predict, free-form play, PCG provided a structured method for the DM to maintain the flow of the game without hitting a creative block Hobbyist magazines such as *Dragon*, *The Judges Guild Journal*, and *Different Worlds* would provide advice for DMs on how to best manage their campaigns and incorporate randomness (e.g. [23,26,29]). With the move towards digital PCG, books of computer-generated monsters [46,47,48] and DM assistant programs [52] were intended to further ease the creative burden.

5.3 PCG As An Expressive Medium

Creating generative systems can be a reward in itself for artists and designers. Though only anecdotal, many designers of PCG systems have the shared experience of spending hours just hitting the "generate" button to see the next surprising, unusual, or amusing piece of content created by their system. The modularity of tabletop roleplaying systems, combined with hobbyist enthusiasm, fostered hobbyist creation of new generative systems, either customized for their own campaign or made generalizable and published in hobbyist magazines of the time period. With the advent of the personal computer and the hobbyist community surrounding it, many new generative art and PCG systems were created and published in magazines such as *Creative Computing*. For these individuals, the joy of making a generative system was equal or greater than utility of the system they created.

5.4 Computer as Unbiased Agent

The *Trolls and Trollkin* supplement of pre-generated monsters describes its contents as being "created by computer program to completely eliminate bias and error". Similarly, a character generator for *Dungeons &* Dragons is described as providing "programmed character creation --- without human hestitation!" [39]. Though this motivation has now all-but-disappeared in digital PCG, early thinking about PCG was that the computer could provide an unbiased, "pure" interpretation of a game's mechanics. The computer, lacking an imagination and ulterior motive that one can assume the human DM would have, is capable only of blindly producing content for players. The use of computer-generated monsters would allow a DM to make an appeal to the authority of the computer in its obedient following of the game rules, and cast blame for ill consequences for the players on the machine.

5.5 Replacing Humans

Early personal computers often lacked the ability to network with other machines, and interfaces for multiplayer games from a single keyboard were often clumsy. With much early digital PCG involving the porting of tabletop roleplaying systems to the computer, PCG took on a role of allowing individuals to play a game that had previously required a social environment and multiple other human players, including a human willing to manage the entire experience. While PCG research now can go to great pains to point out that it is not attempting to replace a human designer but rather augment it, much of early digital PCG explicitly attempted to rid humans from a game experience by offloading the creative work of running the campaign and generating encounters to the machine.

6. DISCUSSION

Analog games allow us to see many lightweight, human-executable algorithms for generating content, and means for directing randomness either via manipulating the content being assembled or the algorithm producing it. Algorithms for generating content in analog games are relatively simple by necessity—just as important as the ability for the system to generate content is that the rules for doing so be interpretable and executable by the player.

6.1 Randomness

These algorithms often adopt randomness in a form that is accessible to a player—the roll of the dice, blindly drawing tiles from a bag, playing with a shuffled deck of cards. Randomness has existed in analog games for millennia, in the form of cards and dice [5]. Randomness alone is typically not thought to be a sufficient qualification for PCG; there must be some content selection, at a minimum, to be directed by the randomly chosen numbers.

However, considering randomness as a core element of PCG in non-digital games lets us examine some of the border cases for what constitutes "content" and "procedurality". Should the generation of a new game board in a single-player *Solitaire* game be considered PCG? In many ways it meets the commonly understood definition of PCG as being the algorithmic creation of game content. The algorithm followed is remarkably simple, but the game board that is created undeniably shapes the player's experience, and the randomness provides replayability across sessions.

The digital equivalent of such games is a game like *Bejeweled* [28], where there is a randomly generated board of simple gems for each session: a tableau of gems is initially laid out, then randomly[1] replaced with new gems when existing gems are removed. Such a type of game content has never, to the author's knowledge, been considered procedurally generated; yet, other games that use very slightly more sophisticated pieces of content that are randomly placed together (e.g. *Robot Unicorn Attack* [1]) are cited as examples of very simple content generation or content selection.

Where lies the distinction? Games that are considered to have this form of simple content generation have an additional layer of indirection, where the pieces being randomly assembled are more complex than the random number generator itself. This complexity arises from human authorship: the "experiential chunks" [34]. In *Robot Unicorn Attack*, though randomly pieced together, were originally created by a human. These pieces were designed to fit together in any order, thus permitting a simpler form of assembly. The final product of the algorithm is a level that offers an illusion of authorship. Yet the underlying algorithm shares more in common with a simple game of *Yahtzee* [24] than the process followed by a human to design a carefully crafted level for *Sonic the Hedgehog* [38].

6.2 A Stagnation of Motivation

Today, research in PCG focuses on more computationally complex algorithmic processes, and is moving away from the modular, random approaches found in analog PCG. Yet the results of the work are quite similar to these early examples: the motivations for the work have remained largely the same. Early motivation and purpose for using PCG in analog games is almost perfectly mirrored in the commonly-cited contemporary motivations. Technical approaches to rapid creation of complex content that obeys design guarantees have drastically improved since the days of random lookup tables (where players are asked to adapt if the content generated is invalid!), but ultimately PCG research is attempting to solve the same set of problems as the early systems themselves were: providing meaningfully replayable experiences, reducing an authoring burden on players and/or designers, and providing content that adapts to the player's current skills. There is still much work to be done towards understanding and addressing these issues; however, there is also room for new goals for PCG that can create new kinds of playable experiences and design tools [11,36].

6.3 Striving for Meaning

The use of randomness to drive encounters, item generation, creature generation, and dungeon generation in tabletop roleplaying systems have been incredibly influential on technical PCG research. Yet at the time these systems were created, designers and players alike critiqued the heavy focus on rules to the exclusion of player experience. In articles and letters to magazines and newsletters, it is possible to see echoes of the calls against formalism or proceduralism seen in game studies today [31]. Game designer Lewis Pulsifer calls for DMs to "Make monsters, not monstrosities"—referring to the core statistics underlying a monster as a "monstrosity" that lacks relevance or meaning to players, and advocating for avoidance of random monster generation altogether:

> *…avoid random aggregations of statistics. If you have several dozen charts and tables…you may come up with a good monster. But, like the proverbial monkeys typing randomly, you'll wait a long time for a Shakespearean-quality creature.* [29]

In an article in *Judges Guild Magazine*, Ronald Pehr argues:

> *There is a distressing tendency on the part of most DMs to treat all encounters between players and monsters on a mechanistic basis… Monsters/NPCs are not just collections of hit dice, armor class, and movement. They have personalities, motives, alignments, and in some cases intelligence exceeding that of the players' characters.* [27]

When these pen and paper systems were made computational, any ability for providing meaning in the generated characters or consideration for player experience was lost. This is a problem still faced in digital PCG today: it is simple to create a vast variety of content, but harder to create *meaningful* content or to understand the qualities of generated content in terms of player experience.

7. CONCLUSION

Understanding the "foundations of digital games" lies, in large part, in understanding the non-digital games that came before. Such games have been drastically influential on the artifacts and processes we research in the digital games community. This paper has presented an overview of approaches to PCG in analog games and argued for their role as precursors for digital PCG. The paper has focused largely on games from North America; future work involves looking more broadly at a wider variety of games and more diverse game design practices.

Seeing PCG as being broader than the computer generation of content towards producing stronger artificial intelligence algorithms, and instead viewing it as a designed system that can be executed by either human or computer, has helped identify new directions for digital PCG research in terms of finding new motivations, designing procedurally authored experiences, and creating PCG systems that have an understanding of meaning and player experience.

8. ACKNOWLEDGMENTS

This paper is based on archival research performed at the *Strong National Museum of Play*, Rochester, NY in the Summer of 2014, supported by a research fellowship offered through the museum. My thanks especially to the library staff for their generous and patient assistance Thanks also to participants in the ITU Symposium on the Future of Procedural Content Generation (November 2014) and the Schloss Dagstuhl Seminar on Computational Intelligence and AI in Games (January 2015) for many fruitful conversations that led to some of the ideas in this paper.

9. REFERENCES

1. [adult swim games]. *Robot Unicorn Attack (PC Game)*. 2010.
2. Altmayer, N. Music Composition: A Different Approach. *5*, 1979.
3. Atari. *Touch Me (Arcade Game)*. 1974.
4. Avalon Hill. *Betrayal at House on the Hill*. 2004.

[1] Note that the exact algorithm for choosing new gems in *Bejeweled* is unknown, and is being assumed (for the sake of example) to be a simple random selection. Indeed, this is one of the interesting properties of digital PCG (and challenges of analyzing it): unlike analog procedural systems that make their rules explicit for players to execute, digital PCG algorithms are tied up in hidden source code.

5. Botermans, J. *The Book of Games: Strategy, Tactics & History*. Sterling Publishing, New York, 2008.

6. Braben, D. and Bell, I. *Elite (BBC Micro)*. Acornsoft, 1984.

7. Carlson, L. Chess Takes a Quantum Leap. *The Sun Chronicle*, 1984, 15.

8. Cook, D. and Marsh, S., eds. *Dungeons & Dragons Fantasy Adventure Game: Expert Rulebook*. TSR Hobbies Inc, 1981.

9. Doull, A. The Death of the Level Designer: Procedural Content Generation in Games. *ASCII Dreams: A Roguelike Developer's Diary*, 2008. http://roguelikedeveloper.blogspot.com/2008/01/death-of-level-designer-procedural.html.

10. Dunnigan, J. and Avalon Hill. *Outdoor Survival (game)*. 1972.

11. Eladhari, M.P., Sullivan, A., Smith, G., and McCoy, J. *AI-Based Game Design: Enabling New Playable Experiences*. UC Santa Cruz Baskin School of Engineering, Santa Cruz, CA, 2011.

12. Emmerich, P. Haiku Generator. *Creative Computing 2*, 1976.

13. Gygax, G. Solo Dungeon Adventures. *The Strategic Review 1*, 1975.

14. Gygax, G. *Advanced Dungeons and Dragons: Players Handbook*. TSR Hobbies, New York, 1978.

15. Gygax, G. and Arneson, D. *Dungeons & Dragons: Rules for Fantastic Medieval Wargames Campaigns Playable with Paper and Pencil and Miniature Figures*. TSR Hobbies, 1974.

16. Hasbro. *Pictionary*. 1994.

17. Hello Games. *No Man's Sky (PS4 Game)*. 2015.

18. Hendrikx, M., Meijer, S., Van der Velden, J., and Iosup, A. Procedural Content Generation for Games: A Survey. *ACM Transactions on Multimedia Computing, Communications and Applications*, (2011).

19. Higinbotham, W. *Tennis for Two (Analog computer game)*. Brookhaven National Laboratory, Brookhaven, NY, 1958.

20. Kazemi, D. Spelunky's Procedural Space. *Tiny Subversions*, 2009. http://tinysubversions.com/2009/09/spelunkys-procedural-space/.

21. Kline, D. and Hetu, L. AI of Darkspore (Invited Talk). *2011 Conference on Artificial Intelligence in Interactive Digital Entertainment (Palo Alto, CA)*, 2011. http://dankline.files.wordpress.com/2011/10/ai-in-darkspore-aiide-2011.pptx.

22. Liapis, A., Yannakakis, G.N., and Togelius, J. Sentient sketchbook: Computer-aided game level authoring. *Proceedings of ACM Conference on Foundations of Digital Games*, (2013).

23. Mayeau, M. Random Monster Tables. *Judges Guild Journal*, 1979, 10–38.

24. Milton Bradley. *Yahtzee (Game)*. 1940s.

25. Milton Bradley. *Simon*. 1978.

26. Olson, E. Random Magic. *Judges Guild Journal*, 1979, 19–26.

27. Pehr, R. Encounters in D&D. *Judges Guild Journal*, 1979, 18–23.

28. PopCap Games. *Bejeweled (PC Game)*. 2001.

29. Pulsifer, L. Make Monsters, Not Monstrosities. *Dragon VI*, 1982, 62–66.

30. Sackson, S. *Cross Currents (Game)*. 1989.

31. Sicart, M. Against Procedurality. *Game Studies 11*, 3 (2011).

32. Smelik, R.M., Tutenel, T., de Kraker, K.J., and Bidarra, R. Integrating Procedural Generation and Manual Editing of Virtual Worlds. *Proceedings of the 2010 Workshop on Procedural Content Generation in Games (co-located with FDG 2010)*, (2010).

33. Smith, A.M. and Mateas, M. Answer Set Programming for Procedural Content Generation: A Design Space Approach. *Computational Intelligence and AI in Games, IEEE Transactions on 3*, 3 (2011), 187 –200.

34. Smith, G. Understanding Procedural Content Generation: A Design-Centric Analysis of the Role of PCG in Games. *Proceedings of the 2014 ACM Conference on Computer-Human Interaction*, (2014).

35. Smith, G., Whitehead, J., and Mateas, M. Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design. *IEEE Transactions on Computational Intelligence and AI in Games (TCIAIG), Special Issue on Procedural Content Generation 3*, 3 (2011).

36. Smith, G., Whitehead, J., and Mateas, M. Computers as Design Collaborators: Interacting with Mixed-Initiative Tools. *Proceedings of the Workshop on Semi-Automated Creativity, co-located with ACM Creativity & Cognition 2011*, (2011).

37. Smith, S. *Adventure Construction Set (Game)*. Electronic Arts, 1984.

38. Sonic Team. *Sonic the Hedgehog (Genesis)*. SEGA, 1991.

39. Spann, J. What do you get when you cross a Dungeon Master with a computer? *Dragon VII*, 1983, 42–48.

40. St. Andre, K. *Tunnels & Trolls*. Flying Buffalo, 1975.

41. The Quantum Game Company. *Quantum: The Game of Modern Life (Game)*. 1984.

42. Togelius, J., Kastbjerg, E., Schedl, D., and Yannakakis, G.N. What is procedural content generation?: Mario on the borderline. *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, (2011), 3.

43. Togelius, J., Yannakakis, G.N., Stanley, K.O., and Browne, C. Search-Based Procedural Content Generation: A Taxonomy and Survey. *Computational Intelligence and AI in Games, IEEE Transactions on 3*, 3 (2011), 172 –186.

44. Toy, M., Wichman, G., Arnold, K., and Lane, J. *Rogue (PC Game)*. 1980.

45. TSR Hobbies Inc. *Dungeons & Dragons: Dungeon Geomorphs*. 1976.

46. Turney, R. *Creatures of Chaos*. Chaosium, 1978.

47. Turney, R. *Trolls and Trollkin*. Chaosium, 1978.

48. Turney, R. *Militia & Mercenaries*. Chaosium, 1979.

49. Wrede, K.-J. and Rio Grande Games. *Carcassonne*. 2000.

50. Z-Man Games. *Tales of the Arabian Nights (Board Game)*. .

51. *A Computer Generated Dungeon*. Flying Buffalo, 1977.

52. *Advanced Dungeons & Dragons Dungeon Masters Assistant, Volume 1: Encounters*. Strategic Simulations Inc, 1988.

53. Ralph H. Baer papers. *Brian Sutton-Smith Library and Archives of Play at The Strong*, .